

An HVS-inspired video deinterlacer based on visual saliency

Umang Aggarwal¹ · Maria Trocan¹ · Francois-Xavier Coudoux²

Received: 14 May 2016 / Accepted: 29 August 2016 / Published online: 16 September 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract Video deinterlacing is a technique wherein the interlaced video format is converted into progressive scan format for nowadays display devices. In this paper, a spatial saliency-guided motion-compensated deinterlacing method is proposed which accounts for the properties of the Human Visual System (HVS): our algorithm classifies the field according to its texture and viewer's region of interest and adapts the motion estimation and compensation, as well as the saliency-guided interpolation to ensure high-quality frame reconstruction. Two different saliency models, namely the graph-based visual saliency (GBVS) model and the spectral residual visual saliency (SRVS) model, have been studied and compared in terms of visual quality performances as well as computational complexity. The experimental results on a great variety of video test sequences show significant improvement of reconstructed video quality with the GBVS-based proposed method compared to classical motion-compensated and adaptive deinterlacing techniques, with up to 4.5 dB gains in terms of PSNR. Simulations also show that the SRVS-based deinterlacing process can result to significant reductions of complexity (up to 25 times a decrease of the computation time compared with the GBVS-based method) at the expense of a PSNR decrease.

Keywords Deinterlacing · Visual saliency · Human visual system (HVS) · Video quality

✉ Maria Trocan
maria.trocan@isep.fr

Francois-Xavier Coudoux
francois-xavier.coudoux@univ-valenciennes.fr

¹ Institut Supérieur d'Electronique de Paris, 28 Rue Notre Dame des Champs, Paris, France

² IEMN (UMR CNRS 8520) Department OAE, Valenciennes University, 59313 Valenciennes Cedex 9, France

1 Introduction

The process of deinterlacing involves converting a stream of interlaced frames within a video sequence to progressive frames [1], to ensure their playback on nowadays progressive devices. Such video processing has been widely studied in the recent literature [2–8], as the interlaced video format is still preferred for the acquisition systems when high-fidelity motion accuracy is needed. Deinterlacing requires the display device to buffer one or more fields and recombine them to a full progressive frame. There are various methods to deinterlace a video and each method produces its own artifacts, due to the temporal lack of information and the dynamics of the video sequence.

Spatial deinterlacers [2,4,7,9] use the information from the current field to interpolate the missing field lines. The most common types of spatial deinterlacing methods are line averaging and directional spatial interpolation. Edge-based line averaging is done by interpolation along the edge direction, by comparing the gradients of various directions. The interpolation accuracy of edge-based line averaging is increased by an efficient estimation of the directional spatial correlations of neighboring pixels. Usually, the spatial deinterlacing methods have low computational power.

However, one disadvantage of spatial deinterlacing is that this class of methods is not optimal due to the fact that motion activity is not considered in interpolation; moreover, these kind of algorithms fail to remove the flickering artifacts.

Motion adaptive methods, such the ones proposed in [5,6,8], use consecutive fields to analyze the characteristics of motion in order to choose the appropriate interpolation scheme. In such deinterlacers, dynamic areas are interpolated spatially and the static segments are interpolated temporally. The best class of deinterlacers is given by the motion-compensated ones [3,10]. In these schemes, the

motion trajectory is estimated and the interpolation of the missing fields is done along the motion flow. However, motion-compensated deinterlacers need massive computational resources. To reduce their complexity, block-based motion estimation is used at the expense of blocking artifacts and some unreliable motion information [11], which severely degrades the visual quality of the reconstructed video sequences.

A single-field interpolation algorithm based on block-wise autoregression that considers mutual influence between the missing high-resolution pixels and the given interlaced, low-resolution pixels in a slip window is introduced in [4]. A method to use different interpolation techniques based on classification of each missing pixel into two categories according to different local region gradient features is discussed in [5]. Further, a statistical-based approach which uses Bayes theory to model the residual of the images as Gaussian and Laplacian distribution can be used to estimate the missing pixels in [6]. To improve the accuracy of motion vectors for video deinterlacing by selectively using optical flow results, for assisting the block-based motion estimation is proposed in [12], at a high computational cost. The computational load of block-based compensation can be reduced using predictive area search algorithms, which estimate the motion vectors (MV) of the current block using the MVs of previous blocks [13]. Neural networks and fuzzy logic can also be used as deinterlacing solutions. A way to exploit fuzzy reasoning to reinforce contours for improving an edge-adaptive deinterlacing algorithm without an excessive increase in computational complexity is discussed in [14]. Another approach for fuzzy logic deinterlacing is to use a fuzzy-bilateral filtering method which considers the range and domain filters based on a fuzzy metric [2,8].

In this paper, in order to reduce the blocking artifacts hence improving the Quality of Experience (QoE) of human viewers, we propose to use the block-based motion estimation on smooth areas, while on highly textured areas optical flow based pixel velocity is used [15] because this method is free of blocking effect. For improving the frame reconstruction quality, visual saliency-guided interpolation of the estimated temporal field is used. The use of visual saliency [16] as trigger for the spatio-temporal interpolator has two advantages: for non-salient regions, no motion estimation is performed, the areas being spatially interpolated, hence highly reducing the proposed deinterlacer complexity.

The second advantage is the corollary of the first one: the computing resources, translated mainly into the motion estimation process, can be used entirely on the region-of-interest area.

In the sequel, the paper is organized as follows: Sect. 2 first introduces the notion of visual saliency and present some existing saliency models. In particular, a focus is made on the graph-based visual saliency (GBVS) model that outperforms

the reference model, as well as the spectral residual visual saliency (SRVS) model. Then, Sect. 3 describes the proposed saliency spatio-temporal video deinterlacing method. Some experimental results obtained with the proposed method for different video sequences are presented in Sect. 4. A comparison between deinterlacing processes using different saliency models is also proposed. Simulation results are presented and discussed. Finally, conclusions are drawn in Sect. 5.

2 Visual saliency

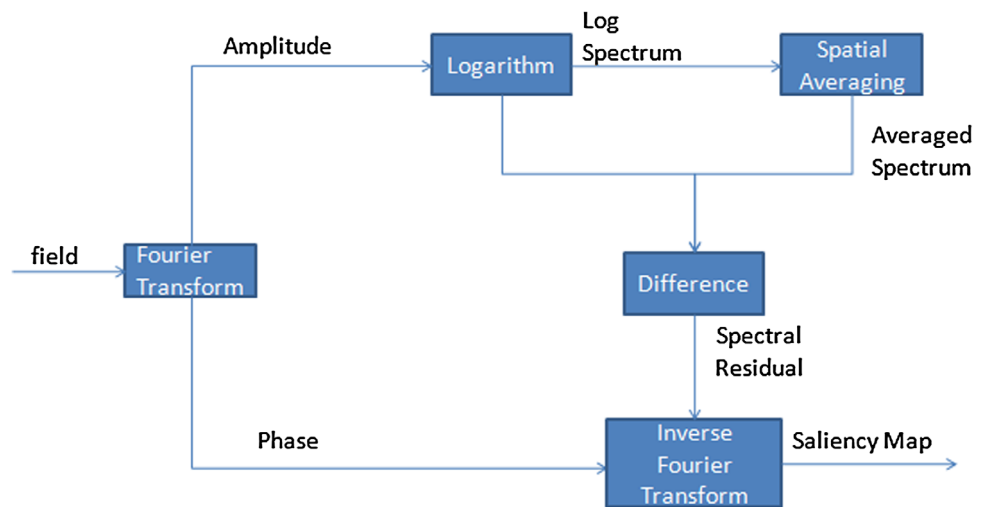
Visual saliency is defined in [17] as *the distinct subjective perceptual quality which makes some items in the world stand out from their neighbors and immediately grab our attention*. The visual saliency process allows a human observer to specifically focus her/his attention on one or more visual stimuli into a scene depending on some semantic features like orientation, motion or color.

It constitutes one of the most important properties of the human visual system (HVS) with numerous applications in digital imaging applications including content-aware video coding, segmentation or image resizing [18,19]. To model human visual attention, several visual saliency models have been recently proposed in the literature [20–22]. Generally, these models allow computing a so-called visual saliency map as a topographically arranged map that represents visually salient parts, also called regions of interest (ROI), of a visual scene. Among the different existing saliency models, the one proposed by Itti et al. [17,23] is the most popular.

The Itti algorithm exploits three low-level semantic features of an image: color, orientation and intensity. These features are extracted from the image to establish feature maps. Finally, the saliency map is computed from these feature maps after normalization and pooling.

In [16], the authors propose a Graph-Based Visual Saliency (GBVS) model which improves the model developed by Itti et al. The GBVS model relies on a fully connected graph between feature maps at multiple spatial scales. It is shown that the GBVS model outperforms the Itti model in predicting human visual attention while viewing natural images. However, the computational complexity of the GBVS model constitutes a significant drawback for deinterlacing implementation purposes. Hence, other low-complex saliency models have been considered to replace the GBVS one.

Among these, we retain the so-called spectral residual visual saliency (SRVS) model described in [10,15]. Figure 1 represents the flowchart of the spectral residual saliency model's computation. The model relies on spectral residual saliency detection. Spectral residual saliency detection is an approach developed in computer vision to simulate the behavior of pre-attentive visual search. Different from

Fig. 1 Flowchart diagram of the spectral saliency model

traditional image statistical models, it analyzes the log spectrum of each image of the video sequence and estimates the corresponding spectral residual. Then, the spectral residual is transformed to the spatial domain to obtain the saliency map. This method explores the properties of the background areas, rather than the target objects. The procedure can be detailed as follows.

Given the luminance component (Y) of a field, the amplitude spectrum noted $A(f)$ and the phase spectrum noted $P(f)$ are first evaluated as the real and the imaginary part of the two-dimensional Fourier transform of the luminance component, respectively:

$$A(f) = \sqrt{\text{Re}[F(Y)]^2 + \text{Im}[F(Y)]^2} \quad (1)$$

$$P(f) = \tan^{-1} \frac{\text{Im}[F(Y)]}{\text{Re}[F(Y)]} \quad (2)$$

where F represents the Fourier transform. The log spectrum $L(f)$ is then obtained by:

$$L(f) = \log[A(f)] \quad (3)$$

The average spectrum can be approximated by convoluting the log spectrum with a matrix $h_n(f)$:

$$A_s(f) = h_n(f) * L(f) \quad (4)$$

where $h_n(f)$ is a $n \times n$ unit matrix with all entries equal to $1/n^2$.

Finally, the spectral residual $R(f)$ consists in the statistical singularities specific to the input image and is obtained, for each frame of a video sequence, as the difference between the log spectrum and the averaged spectrum, respectively:

$$R(f) = L(f) - A_s(f) \quad (5)$$

Spectral residual is then converted to the saliency map $S(x)$ using inverse two-dimensional Fourier transform. The resulting saliency map contains primarily the non-trivial part of the visual scene. The value at each point in a saliency map is squared to indicate the estimation error. For better visual effects, the saliency map is traditionally smoothed with a Gaussian filter $g(x)$ with typical variance of 8:

$$S(x) = g(x) * F^{-1} \left[\exp(P(f) + R(f))^2 \right]. \quad (6)$$

3 Saliency-based deinterlacing

The flowchart of the proposed algorithm is depicted in Fig. 2. As the field interpolation model depends on the saliency map, the first step of our algorithm is given by the computation of the spatial saliency of the current field to the deinterlaced. To compute the so-called saliency map, the two models described in the Sect. 2 have been considered: the GBVS model proposed in [16], and the spectral saliency model. The obtained saliency map denoted in the followings by S (i.e., depicted in Fig. 3 in the case of the GBVS model) and consisting of gray values $S(i, j) \in \{0 \dots 255\}$ will trigger, along with the texture type, the interpolation used for the current field. Equally, a Canny edge detector is applied on the current field and the edges mask C is obtained.

Further, the current field is partitioned into blocks of fixed size B^2 , each block being categorized depending on its belonging to the salient region, as follows: the block b_n of size B^2 is said to be salient/important, i.e.:

$$S_{b_n} = \sum_{i=1}^B \sum_{j=1}^B s_n(i, j) / B^2 \quad (7)$$

if the mean S_{b_n} of the entire collocated block s_n within the saliency map S is higher than a given threshold T_s ; otherwise, the block is classified as smooth.

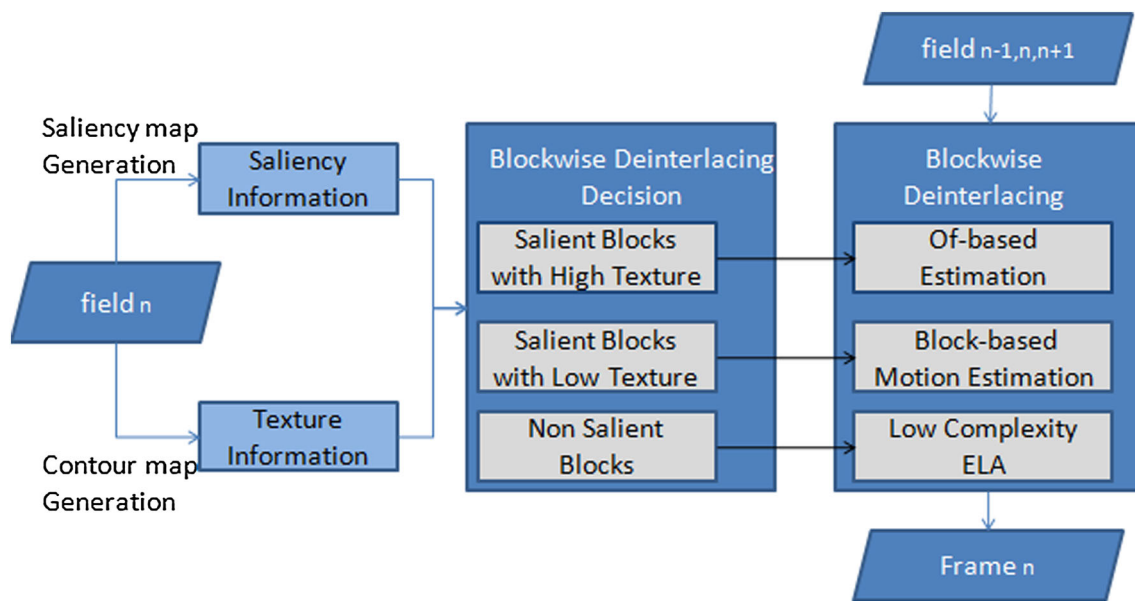


Fig. 2 Flowchart diagram of the proposed deinterlacing algorithm

Also, for each block b_n belonging to the current field f_n , its number of edges is derived as in Eq. (8), by counting the amount of pixels on contours in the collocated block c_n in the mask field C , obtained with the Canny filter:

$$CE_{b_n} = \sum_{i=1}^B \sum_{j=1}^B c_n(i, j) \quad (8)$$

where, CE_{b_n} is the number of identified edges in block b_n . The block b_n is classified as highly textured if CE_{b_n} is significant with respect to the blocksize B^2 , i.e.:

$$CE_{b_n} > T_b \quad (9)$$

(T_b is a threshold depending on B^2), or smooth, if Eq. (9) does not hold.

If the block b_n belongs to a salient region and its number of contours is significant (as in Eq. 9), optical flow-based motion estimation is implemented; otherwise, we use block-based estimation.

If the block b_n is determined as not belonging to a salient region, simple spatial 5-tap edge-line averaging techniques (Fig. 4) are used to obtain the deinterlaced block $\hat{b}_n(i, j)$, i.e.:

$$\hat{b}_n(i, j) = \frac{b_n(i-1, j+x_0) + b_n(i+1, j-x_0)}{2}, \quad (10)$$

where the exact value of x_0 is given by the minimization:

$$\begin{aligned} & |b_n(i-1, j+x_0) - b_n(i+1, j-x_0)| \\ &= \min_{x_0 \in \{-2, -1, 0, 1, 2\}} |b_n(i-1, j+x_0) - b_n(i+1, j-x_0)|. \end{aligned} \quad (11)$$

For the salient blocks, the motion vectors (MV) are obtained on the backward and forward directions for the current field, and applying either OF-based estimation proposed by Liu in [5], or simple block-based ME.

We assume that the motion trajectory is linear; so, the obtained forward motion vectors (MVs) are split into backward (MVB) and forward (MVF) motion vector fields for the current field f_n . As a block in f_n could have zero or more than one MVs passing through, the corresponding MV_n for the block $b_n \in f_n$ is obtained by the minimization of the Euclidean distance between b_n 's center, $(y_{n,0}, x_{n,0})$, and the passing vectors MVs. In our minimization, we consider only the MVs obtained for the blocks in the neighborhood of the collocated block b_{n-1} in the left field f_{n-1} (thus, a total of nine MVs, obtained for b_{n-1} and the blocks adjacent to $b_{n-1} \in f_{n-1}$, as these MVs are supposed to be the most correlated to the one in the current block, e.g., belonging to the same motion object).

If the motion vector MV corresponding to the collocated block $b_{n-1} \in f_{n-1}$ lies on the line:

$$\frac{y - y_{n-1,0}}{MV_y} = \frac{x - x_{n-1,0}}{MV_x} \quad (12)$$

where $(y_{n-1,0}, x_{n-1,0})$ is the center of b_{n-1} ; and MV_x , respectively, MV_y , measures the displacement along the x , respectively, y axis; the distances from the center $(y_{n,0}, x_{n,0})$ of the current block b_n to the MVs lines are obtained as:



Fig. 3 Saliency map obtained for (a) 10th frame of “Foreman” sequence, (b) 21st frame of “Salesman” sequence

$$D_{k \in \{1, \dots, 9\}} = \frac{|MV_{k,x}y_{n,0} - MV_{k,y}x_{n,0} + MV_{k,y}x_{n-1,k} - MV_{k,x}y_{n-1,k}|}{\sqrt{MV_{k,x}^2 + MV_{k,y}^2}} \quad (13)$$

MV^n is the closest motion vector to the current block b_n , if its corresponding distance to the center of b_n , $(y_{n,0}, x_{n,0})$, is minimal, i.e., $D_n = \min(D_{k \in \{1, \dots, 9\}})$. Hence, MV^n is generated for each block, containing the motion estimation in the x and y directions for every pixel.

The forward and backward MVs for each block are obtained as:

$$MV_B^n = \frac{-MV^n}{2}, \quad MV_F^n = \frac{MV^n}{2}. \quad (14)$$

The backward prediction of b_n , denoted by $\mathcal{F}_{MV_B}^n$, is obtained as:

$$\mathcal{F}_{MV_B}^n(i, j) = b_{n-1}(i + MV_{By}^n, j + MV_{Bx}^n), \quad (15)$$

and the forward prediction of the current block, $\mathcal{F}_{MV_F}^n$, is obtained as:

$$\mathcal{F}_{MV_F}^n(i, j) = b_{n+1}(i + MV_{Fy}^n, j + MV_{Fx}^n), \quad (16)$$

The motion-compensated block \hat{b}^n to be further used for the deinterlacing of b_n is obtained as average of the backward, $\mathcal{F}_{MV_B}^n$, and forward, $\mathcal{F}_{MV_F}^n$, predictions:

$$\hat{b}^n(i, j) = \frac{\mathcal{F}_{MV_B}^n(i, j) + \mathcal{F}_{MV_F}^n(i, j)}{2}. \quad (17)$$

Finally, the deinterlaced block is found in a saliency-based motion-compensated manner, as:

$$\hat{b}_n(i, j) = \frac{b_n(i-1, j+x_0) + b_n(i+1, j-x_0) + s_n(i, j)\hat{b}^n(i, j)}{s_n(i, j) + 2}, \quad (18)$$

s_n being the corresponding saliency value within the saliency map S , acting as a weight for the motion-compensated interpolation, and x_0 is obtained by the edge line minimization in (11).

4 Experimental results

To objectively and comprehensively present the performance of the proposed deinterlacing approach, our method has been tested on several CIF- 352×288 (“Foreman”, “Hall”, “Mobile”, “Stefan” and “News”) and QCIF- 176×144 (“Carphone” and “Salesman”) video sequences. These well-known test sequences have been chosen for their different texture content and motion dynamics. Such spatio-temporal characteristics can be explicated by computing the relative spatial information (SI) and temporal information (TI) found in these video contents, as described in [24]. Figure 5 shows the relative amount of spatial and temporal information for the selected test scenes. We can note that they span a large portion of the SI–TI plane, as desired. Moreover, the two (SI, TI) pairs located in the top right part of the diagram correspond to the “Mobile” and “Stefan” CIF sequences which are known to contain very high spatio-temporal activity.

The selected video sequences were originally in progressive format. To generate interlaced content, the even lines of the even frames and the odd lines of the odd frames were removed, as shown in Fig. 6. This way, objective quality measurements could be done, using the original sequences—progressive frames—as references.

Fig. 4 5-Tap Edge Line Average (ELA): the five interpolation directions are represented with different dashed lines. Gray nodes correspond to original pixels from the upper and lower lines (solid lines); the black node is the interpolated pixel

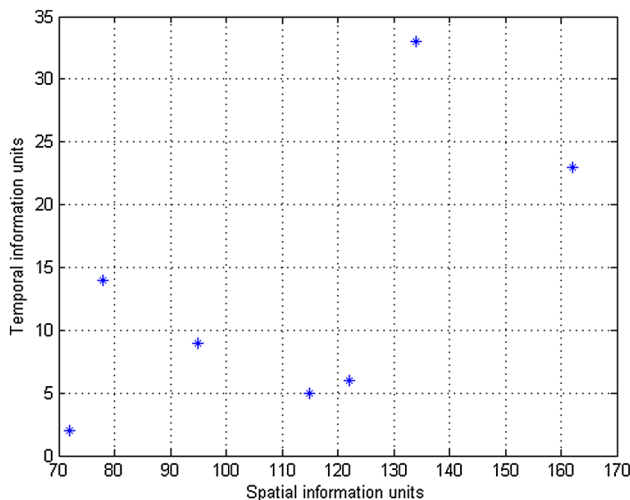
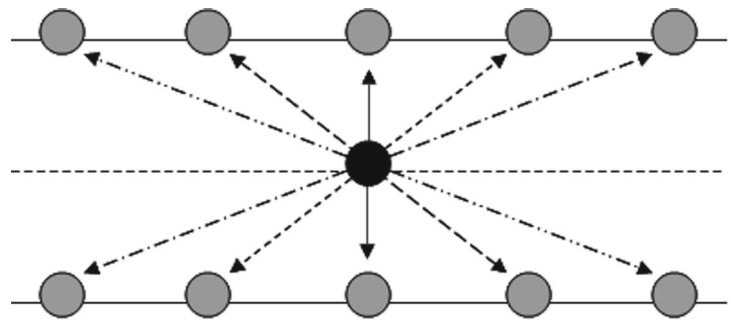


Fig. 5 Spatial-temporal information diagram for the test video scene set

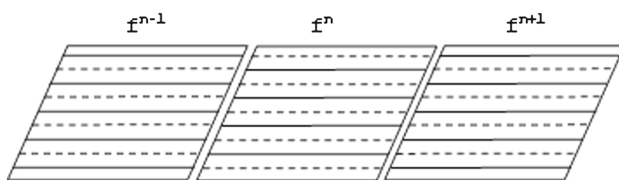


Fig. 6 Progressive to interlaced format frame conversion, by removing the dashed lines

In our experimental framework, the GBVS model is first considered. We have used 8×8 ($B = 8$) pixel blocks for a 16×16 ($S = 16$) search motion estimation window, for the salient blocks b_n having a small $CE_{b_n} < T_b$ number of contours. The parameter T_s for saliency detector was set up to 20, and the edge threshold T_b to 32 (e.g., at least half of the block pixels are situated on contours).

The tests were run on 50 frames for each sequence. The deinterlacing performance of our method is presented in terms of peak signal-to-noise ratio (PSNR) computed on the luminance component. The efficiency of our proposed method—denoted in the followings by SGAD—is compared in Table 1 to Vertical Average (VA), Edge Line Average (ELA), Temporal Field Average (TFA), Adap-

tive Motion Estimation (AME) and Motion-Compensated Deinterlacing (MCD), which are the most common implementations in deinterlacing systems. Moreover, the proposed algorithm is compared to the work in [27], denoted by EPMC, [28] denoted by SMCD and the methods proposed in [29] (high-fidelity motion estimation based deinterlacer), [30] (adaptive motion-compensated interpolator with overlapped motion estimation) and [31] (hybrid low-complexity motion-compensated-based deinterlacer), which are all motion-compensation-based algorithms with different complexity degrees. (these latter results are reported as in the corresponding references, NC denoting the non-communicated ones). In the present case, the GBVS model is considered.

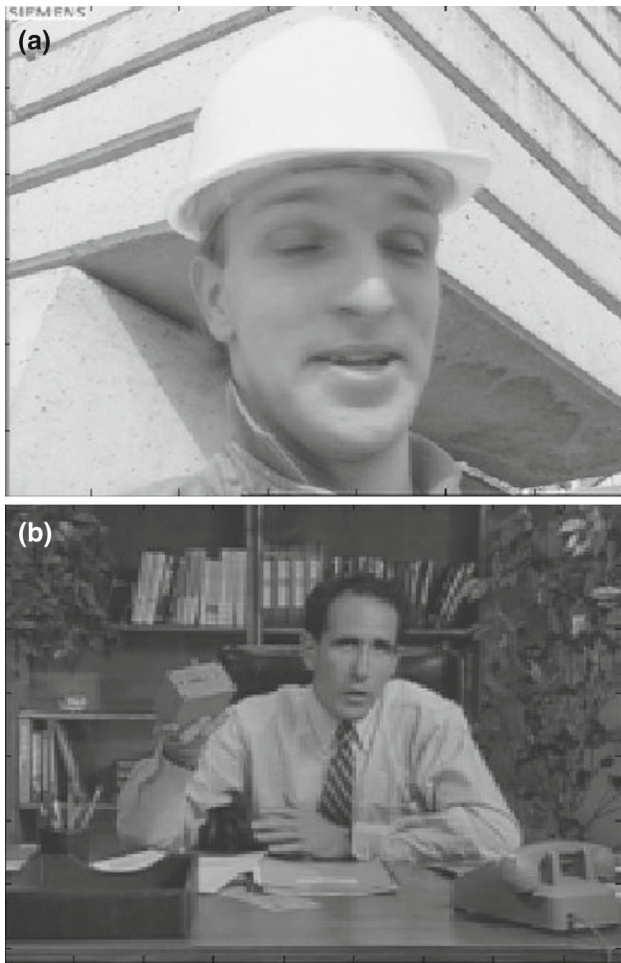
For visually showing the results of the proposed method, two deinterlaced frames are illustrated in Fig. 7.

As it can be seen in the presented results, our proposed method using the GBVS model has an average PSNR gain of ≈ 4.5 dBs with respect to a wide range of deinterlacers. Our framework has been implemented in Matlab (8.0.0.783 (R2012b)) and the tests have been realized on a quad-core Intel-PC@4 GHz. Due to the independent block-based processing, the proposed deinterlacing approach is prone to distributed/parallel implementation, thus highly reducing the computation time obtained with a sequential implementation. Moreover, as the proposed algorithm adapts the motion estimation in function of region's saliency, due to our used threshold T_s for motion computation, only $\approx 1/3$ of field regions is motion processed (as it can be seen in Fig. 3). The parameterization allows, thus, to drastically decrease the complexity attached to motion-compensated schemes, by preserving its advantages where the user attention is focused.

However, it is known that the GBVS model requires a lot of computational resources to be computed. Such computational complexity can be a severe issue particularly for real-time applications. For lower complexity, we propose in what follows to replace the GBVS model by the SRVS one to compute the saliency values. Because optical flow is also complex to implement, only block-based MC is used in the low-complex approach; the implementation of the algorithm is left unchanged. The performances of the low-complex

Table 1 Y-PSNR results (in dB)

	Foreman	Hall	Mobile	Stefan	News	Carphone	Salesman
VA	32.15	28.26	25.38	27.30	34.64	32.17	31.52
ELA	33.14	30.74	23.47	26.04	32.19	32.33	30.51
TFA	34.08	37.47	27.96	26.83	41.06	37.39	45.22
AME	33.19	27.27	20.95	23.84	27.36	29.63	28.24
MCD	35.42	34.23	25.26	27.32	35.49	33.55	33.16
EPMC (S1)	37.09	39.27	31.54	30.02	41.63	37.53	45.61
EPMC (S2)	37.18	39.08	30.56	30.11	39.44	37.55	42.28
[29]	33.77	NC	27.66	28.79	NC	NC	NC
[30]	NC	NC	NC	24.59	NC	NC	NC
[31]	33.93	38.79	24.67	26.88	NC	NC	NC
SMCD (S1)	37.52	39.71	30.41	31.77	41.85	37.59	45.95
SMCD (S2)	37.63	39.86	30.58	31.82	42.00	37.74	45.09
SGAD	39.07	43.86	37.54	34.23	44.35	40.33	50.70

**Fig. 7** Deinterlacing result for the **a** 10th frame of “Foreman” sequence, **b** 21th frame of “Salesman” sequence

SGAD algorithm using the SRVS model are evaluated by comparing the PSNR and computation time values with the GBVS-based method for the video data set. Table 2 summarizes the results that include the average PSNR value, the average number of blocks on which the ELA algorithm is used (noted N_e), the average number of blocks on which block-based MC is used (noted N_m), and the average total computation time noted CT (saliency’s estimation then adaptive block processing).

First, we can note that the average PSNR values are reduced compared to the ones obtained with the GBVS-based method. The quality loss is due to the artifacts introduced by the block-based MC process, as opposed to optical flow mostly, but also due to the saliency model performance: the GBVS model has the best results, but unfortunately at the expense of processing time (it takes about 1 min to extract the saliency map). Nevertheless, we verify that the low-complex algorithm offers performances which are mostly similar to conventional deinterlacing techniques in terms of video quality. Concerning the total processing time, it varies between 4.49 and 20.1 s. This time must be set against that required for the GBVS-based version which varies approximately from 100 s for QCIF video contents to 350–400 s for CIF ones. Such time penalty for the initial version of our algorithm is mainly due to the optical flow computation, especially if highly textured salient regions are represent in the initial scene. Hence, the modified version of the SGAD algorithm can be adapted for real-time deinterlacing while maintaining a satisfactory video quality though slightly reduced. On the contrary, the GBVS version should be better suited for storage for which deinterlacing time is not an issue; so,

Table 2 performances of the modified low-complex SGAD algorithm

	Foreman	Hall	Mobile	Stefan	News	Carphone	Salesman
PSNR (dB)	37.32	36.99	29.49	30.78	41.65	37.81	41.48
N_e (%)	73	8	1	6	9	9	9
N_m (%)	27	92	99	94	91	91	91
CT (s)	6.57	19.68	20.10	19.90	19.16	4.49	4.76

it is mainly designed for adapting the content from interlaced cameras to progressive display devices. To conclude, it should be noted that further gains should be expected for the proposed SGAD method because the code is still not optimized.

5 Conclusion

In this paper, a spatial saliency-guided motion-compensated method for video deinterlacing is proposed. Our approach is an efficient deinterlacing tool, being able to adapt the interpolation method depending both on region of interest and its texture content. Experiments show that the proposed algorithm generates high-quality results, having more than 4.5 dBs PSNR gain, in average, compared to other deinterlacing approaches. Furthermore, the proposed method acknowledges the possibility of improving image quality and simultaneously reducing execution time, based on the saliency map. Finally, we have presented two models: the first one for storage applications (in this case, deinterlacing time is not a critical issue, so it is mainly designed for high-quality conversion from interlaced cameras to progressive display devices), and the other one with less but still acceptable video quality performances, which can be adapted for real-time deinterlacers.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Haan, G.D., Bellers, E.B.: Deinterlacing: an overview. *Proc. IEEE* **86**(9), 1839–1857 (1998)
- Jeon, G., Anisetti, M., Wang, L., Damiani, E.: Locally estimated heterogeneity property and its fuzzy filter application for deinterlacing. *Inf. Sci.* **354**, 112–130 (2016)
- Yang, W.-J., Chung, K.-L., Huang, Y.-H., Lin, L.-C.: Quality-efficient syntax element-based deinterlacing method for H.264-coded video sequences with various resolutions. *J. Vis. Commun. Image R* **25**, 466–477 (2014)
- Wang, J., Jeon, G., Jeong, J.: A block-wise autoregression-based deinterlacing algorithm. *J. Disp. Technol.* **10**(5), 414–419 (2014)
- Zhang, H., Rong, M.: Deinterlacing algorithm using gradient-guided interpolation and weighted average of directional estimation. *IET Image Process.* **9**(6), 450–460 (2015)
- Wang, J., Jeon, G., Jeong, J.: A hybrid algorithm using maximum a posteriori for interlaced to progressive scanning format conversion. *J. Disp. Technol.* **11**(2), 183–192 (2015)
- Abboud, F., Chouzenoux, E., Pesquet, J.-C., Chenot, J.-H., Laborelli, L.: A dual block coordinate proximal algorithm with application to deconvolution of interlaced video sequences. In: *IEEE International Conference on Image Processing*, pp. 4917–4921 (2015)
- Jeon, G., Kang, S., Lee, J.-K.: A robust fuzzy-bilateral filtering method and its application to video deinterlacing. *J. Real-Time Image Proc.* **11**(1), 223–233 (2016)
- Atkins, C.B.: Optical image scaling using pixel classification. In: *International Conference on Image Processing* (2001)
- Liu, C.: Beyond pixels: exploring new representations and applications for motion analysis. Doctoral Thesis, MIT (2009)
- Trocan, M., Mikovicova, B., Zhanguzin, D.: An adaptive motion compensated approach for video deinterlacing. *Multimed. Tools Appl.* **61**(3), 819–837 (2011)
- Wang, C., Huang, R., Miao, W., Zhao, J., He, J.: Video deinterlacing method based on optical flow. In: *IEEE International Conference on Wireless Communications and Signal Processing (WCSP)* (2012)
- Abdoli, B.: A dynamic predictive search algorithm for fast block-based motion estimation. Theses and Dissertations, Digital Library Ryerson Canada (2012)
- Brox, P., Baturone, I., Sanchez-Solano, S., Gutierrez-Rios, J.: Edge-adaptive spatial video deinterlacing algorithms based on fuzzy logic. *IEEE Trans. Consum. Electr.* **60**(3), 375–383 (2014)
- Horn, B.K.P., Schunck, B.G.: Determining optical flow. *Artif. Intell.* **17**, 185203 (1981)
- Harel, J., Koch, C., Pietro, P.: Graph-Based Visual Saliency. In: *Advances in Neural Information Processing Systems* (2006)
- Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. PAMI* **20**(11), 1254–1259 (1998)
- Itti, L.: Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Trans. Image Process.* **13**(10), 1304–1318 (2004)
- Hou, X., Zhang, L.: Saliency detection: a spectral residual approach. *IEEE Conf. Comput. Vis. Pattern Recogn.* **1**, 1–8 (2007)
- Schauerte, B., Stiefelhagen, R.: Quaternion-based spectral saliency detection for eye fixation prediction. *Eur. Conf. Comput. Vis. (ECCV)* **7573**, 116–129 (2012)
- Rahtu, E., Kannala, J., et al.: Segmenting salient objects from images and videos. In: *Proceedings of European Conference on Computer Vision (ECCV2010)*, pp. 321–332 (2010)
- Zhang, L., Tong, M., et al.: SUN: a Bayesian framework for saliency using natural statistics. *J. Vis.* **9**(7), 1–20 (2008)

23. Lu, S., Lim, J.-H.: Saliency modeling from image histograms. In: European Conference on Computer Vision (ECCV), pp. 321–332. Florence (2012)
24. Recommendation ITU-T P. 910: Subjective video quality assessment methods for multimedia applications, pp. 14–17 (2008)
25. Seo, H.-J., Milanfar, P.: Static and space-time visual saliency detection by self-resemblance. *J. Vis.* **9**(12), 1–12 (2009)
26. Itti, L., Koch, C.: Computational modeling of visual attention. *Nat. Rev. Neurosci.* **2**(3), 194–203 (2001)
27. Zhanguzin, D., Trocan, M., Mikovicova, B.: An edge-preserving motion-compensated approach for video deinterlacing. *IEEE/IET/BCS3rd International Workshop on Future Multimedia Networking* (2010)
28. Trocan, M., Mikovicova, B.: Smooth motion compensated video deinterlacing. In: *Image and Signal Processing and Analysis (ISPA), 7th International Symposium* (2011)
29. Chen, Y., Tai, S.: True motion- compensated deinterlacing algorithm. *IEEE Trans. Circ. Syst. Video Technol.* **19**, 1489–1498 (2009)
30. Wang, S.-B., Chang, T.-S.: Adaptive deinterlacing with robust overlapped block motion compensation. *IEEE Trans. Circ. Syst. Video Technol.* **18**(10), 1437–1440 (2008)
31. Lee, G., Wang, M., Li, H., et al.: A motion-adaptive deinterlacer via hybrid motion detection and edge-pattern recognition. *J Image Video Proc* 2008:741290 (2008). doi:[10.1155/2008/741290](https://doi.org/10.1155/2008/741290)